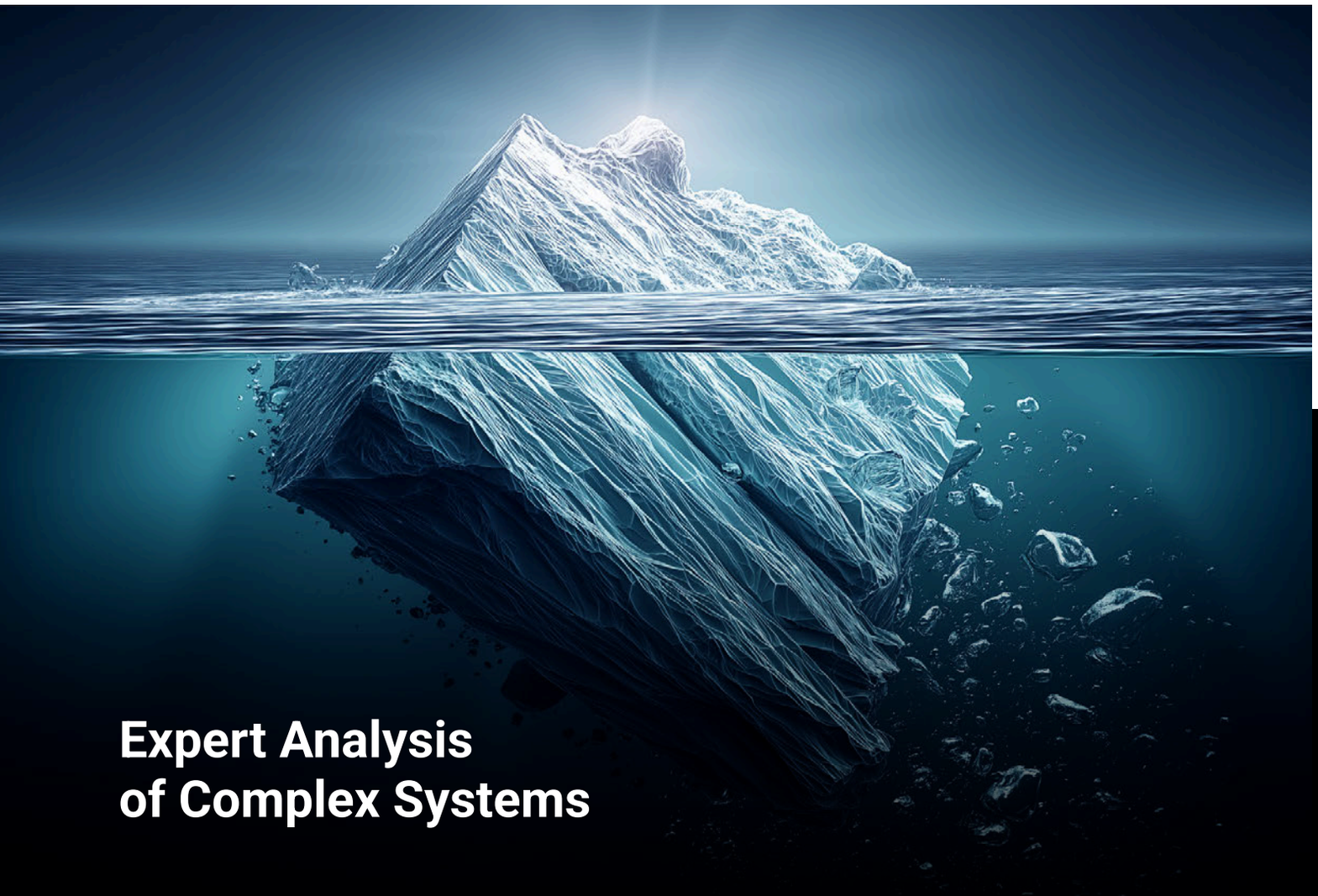
A MULTI-FACETED ANALYSIS

The Unseen Costs and Latent Risks of Open Source Software

PREPARED BY
Mahdi Eslamimehr PhD, MBA



Expert Analysis of Complex Systems

QUANDARY PEAK RESEARCH

THE PEAK OF EXPERTISE

Executive Summary

Open Source Software (OSS) has become the bedrock of modern digital infrastructure, fueling innovation and economic growth with an estimated demand-side value of nearly $9 trillion. Its ubiquity, however, masks a complex and often underestimated landscape of risks spanning economic valuation, security vulnerabilities, legal compliance, and hidden operational costs.

This article presents a comprehensive, multi-faceted analysis of the inherent risks associated with leveraging OSS. Drawing on recent empirical studies and industry reports, we quantify the scale of these challenges, from the 86% of commercial applications containing at least one vulnerability to the multi-million dollar legal penalties for license non-compliance.

We further dissect the Total Cost of Ownership (TCO), revealing significant hidden expenses that belie the "free" nature of OSS. Finally, we trace the evolution of sophisticated software supply chain attacks, highlighting the escalating threat landscape. The findings underscore the critical need for a holistic and proactive risk management framework for any organization building on open source foundations.

Introduction

Open Source Software (OSS) represents a fundamental paradigm shift in software development and distribution, fostering a collaborative ecosystem that has accelerated technological progress at an unprecedented rate. An estimated 70% to 90% of any given modern software package is composed of OSS [1], and 97% of commercial applications contain open source components [2]. This widespread adoption has generated immense economic value, with recent studies placing its demand-side or replacement value at a staggering **$8.8 trillion** [3]. This value is created by the millions of developers who contribute to a shared digital commons, allowing firms and individuals to build upon a vast repository of existing code rather than starting from scratch.

However, the very openness and decentralized nature that make OSS so powerful also introduce a complex web of inherent risks. The "free" price tag of open source is a misnomer, obscuring significant hidden costs and potential liabilities. These risks are not monolithic but are distributed across multiple domains, including:

Software Valuation
The paradox between the low cost of creation (supply-side) and the immense replacement value (demand-side) creates market distortions and misaligned incentives for maintenance and security.

Security Vulnerabilities
The reuse of components across thousands of applications means a single vulnerability can have a cascading, global impact, as evidenced by incidents like Log4Shell.

Legal And Compliance
A complex patchwork of over 200 major OSS licenses creates a legal minefield, where non-compliance can lead to costly litigation, loss of intellectual property, and injunctions on product distribution.

Hidden Costs and TCO
The Total Cost of Ownership extends far beyond initial acquisition to include ongoing maintenance, bug fixing, security patching, and compliance management.

Software Supply Chain Attacks
Malicious actors are increasingly targeting the OSS ecosystem itself, injecting malware into popular libraries to compromise downstream users in highly sophisticated supply chain attacks.

This paper aims to provide a scholarly, data-driven analysis of these interconnected risks. By synthesizing findings from the attached research paper [3], recent industry security audits [1, 2], legal case analyses [4], and total cost of ownership studies [5], we will construct a holistic view of the challenges. The analysis will be supported by a series of data visualizations to quantify the scale of each risk category. The objective is not to discourage the use of OSS, but to foster a more informed and risk-aware approach to its adoption, enabling organizations to harness its immense benefits while mitigating its latent dangers.

A FOUNDATION OF RISK

The Dual Nature of OSS Valuation

The economic structure of Open Source Software is characterized by a profound asymmetry between its creation cost and its value to the global economy. A 2024 study by Hoffmann, Nagle, and Zhou provides a foundational model for understanding this duality, estimating both the supply-side (re-creation) and demand-side (replacement) value of the ecosystem [3]. This economic paradox is the wellspring from which many other risks flow.

Supply-Side vs. Demand-Side Value

The **supply-side value**, representing the cost to recreate the existing body of open source code, is estimated to be **$4.15 billion**. This figure, while substantial, pales in comparison to the **demand-side value**, which is the estimated cost firms would incur if they had to replace the OSS they use with proprietary alternatives. This value is estimated to be **$8.8 trillion** [3].

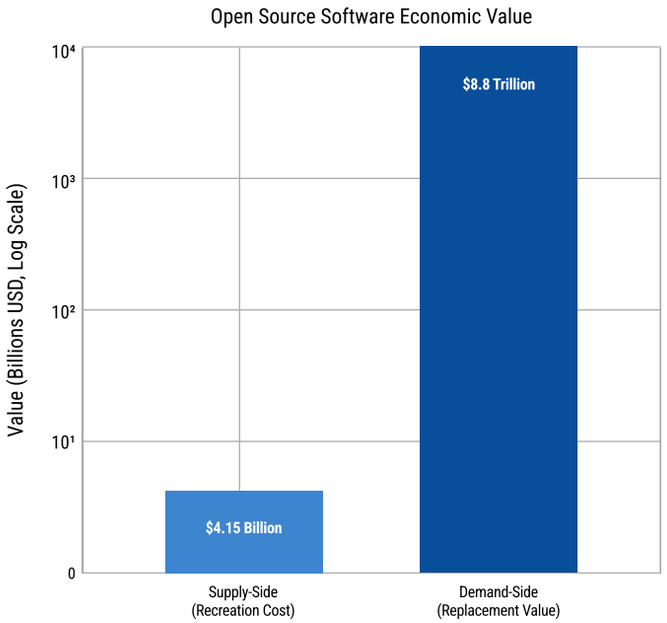


Figure 1: A comparison of the supply-side recreation cost and the demand-side replacement value of the open source ecosystem, highlighting the immense value disparity.
Data Source: Hoffmann, Nagle & Zhou, 2024

This staggering 2000x difference underscores the immense leverage that OSS provides. The study found that without OSS, firms would have to spend **3.5 times** more on proprietary software to achieve the same functionality [3]. However, this leverage comes with a hidden cost: the market provides insufficient incentives to maintain and secure this critical digital infrastructure. The relatively low cost of contribution, combined with the public good nature of the output, leads to underinvestment in the long-term health and security of the ecosystem.

Concentration and Systemic Risk

The value and contribution within the OSS ecosystem are also highly concentrated, creating points of systemic risk. The research indicates that:

- **84%** of the total supply-side value is concentrated in just six programming languages.
- The top **1%** of packages account for **80%** of all package downloads.
- A mere **five individual developers** are responsible for packages that generate **96%** of the total supply-side value [3].

This concentration means that the failure or compromise of a small number of popular packages or the burnout of a handful of key maintainers could have catastrophic, cascading effects across the entire digital economy. The attempted backdoor in the XZ-Utils library in 2024, a project maintained by a single individual, is a stark illustration of this very real threat [6]. The project, a dependency in nearly every major Linux distribution, was nearly compromised by a sophisticated, two-year social engineering attack, an event that would have been a digital apocalypse had it not been discovered by chance.

This valuation paradox and high concentration create a fragile foundation. The immense value derived from OSS is not matched by a proportional investment in its security and sustainability, leaving the entire digital world exposed to significant systemic risks.

SECURITY VULNERABILITIES

A Persistent and Pervasive Threat

The widespread reuse of open source components means that vulnerabilities are not isolated incidents but systemic weaknesses that can be inherited by thousands of downstream applications. The data from recent security audits paints a concerning picture of the prevalence and nature of these vulnerabilities.

Vulnerability Prevalence

According to the 2025 Black Duck Open Source Security and Risk Analysis (OSSRA) report, which analyzed over 1,000 commercial codebases, the vast majority of modern applications are built on a foundation of potentially insecure code [2].

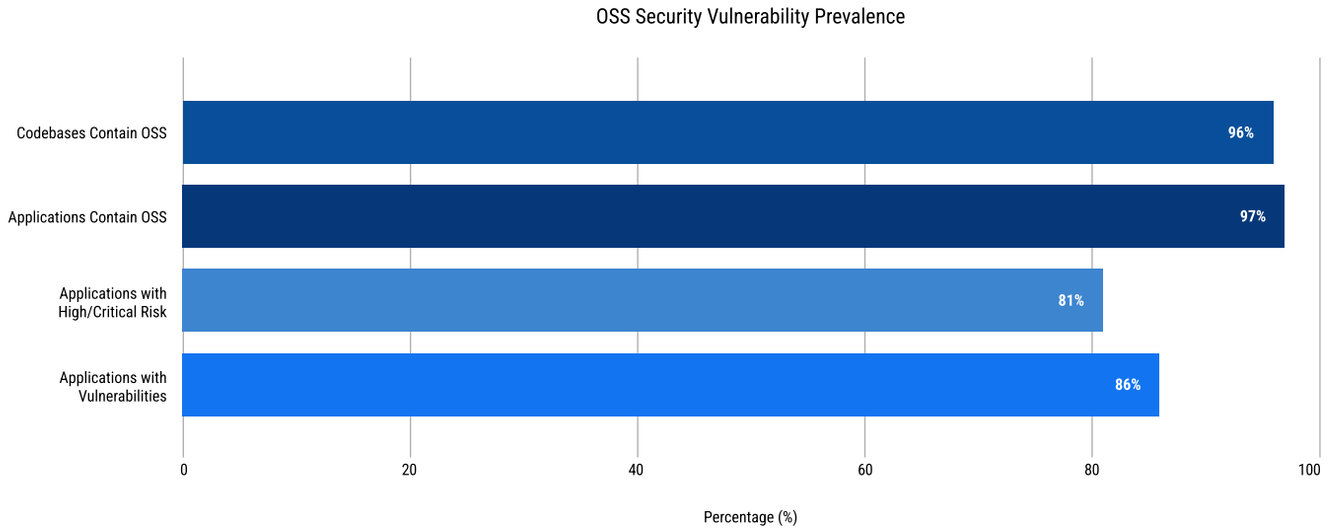


Figure 2: The prevalence of security vulnerabilities in commercial applications highlights the widespread nature of the risk. Data Source: Black Duck OSSRA, 2025

The key findings include:

- **96%** of codebases contained some open source.
- **86%** of the applications contained at least one open source vulnerability.
- **81%** contained at least one vulnerability rated as high or critical.

This data demonstrates that vulnerability exposure is not a rare occurrence but the default state for the majority of software. The problem is exacerbated by poor maintenance hygiene.

The Risk of Outdated Components

A significant portion of the risk comes not from zero-day exploits, but from the failure to apply available patches for known vulnerabilities. The OSSRA report found that:

- 91% **of applications contained outdated versions of open source components.**
- An alarming **90%** of applications were using components that were more than 10 versions out of date [2].

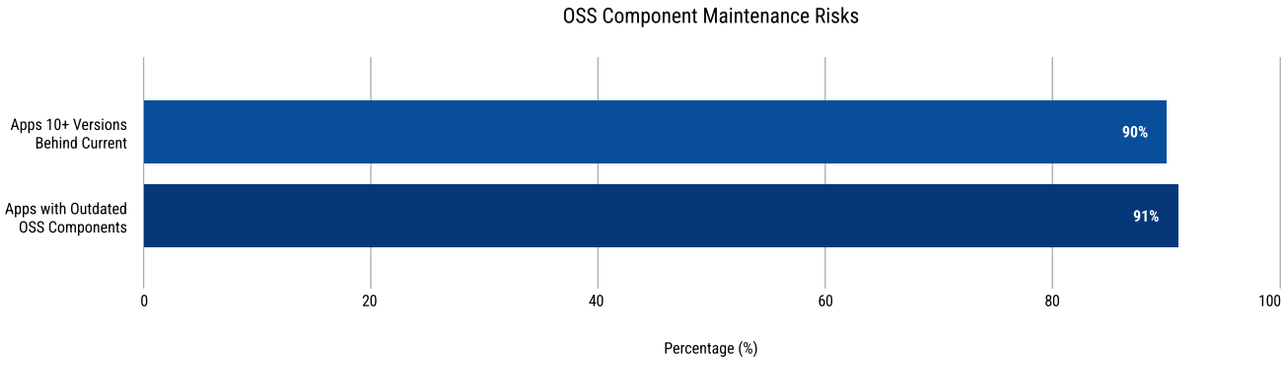


Figure 3: A significant majority of applications rely on outdated OSS components, often by many versions, dramatically increasing the window of opportunity for attackers. Data Source: Black Duck OSSRA, 2025

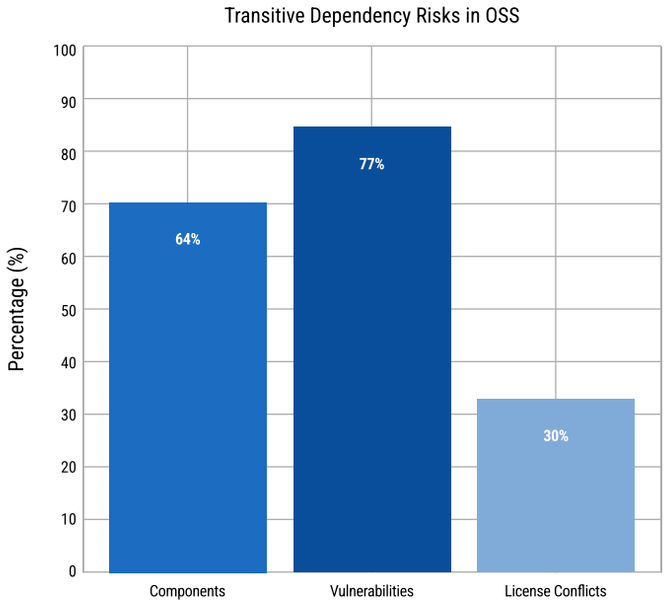
This lack of maintenance creates a massive window of opportunity for attackers, who can exploit well-documented vulnerabilities that have had patches available for months or even years. The 2017 Equifax breach, which was caused by the failure to patch a two-month-old vulnerability in Apache Struts, remains a canonical example of the catastrophic consequences of such negligence [6].

Transitive Dependencies: The Hidden Risk Multiplier

The complexity of the software supply chain means that developers are often unaware of the full extent of their reliance on open source. The dependencies they directly include in their projects often have their own dependencies, known as **transitive dependencies**. These hidden components are a major source of risk. Data shows that:

- **64%** of all open source components in commercial applications are transitive dependencies [2].
- **77%** of all open source vulnerabilities are found in these transitive dependencies [7].

Figure 4: Transitive dependencies constitute the majority of components and are the source for the vast majority of vulnerabilities and license conflicts.
Data Source: Black Duck OSSRA 2025, Lineaje



This "dependency of a dependency" problem makes risk management exponentially more difficult. Organizations may diligently vet their direct dependencies while remaining completely blind to vulnerabilities lurking deeper in the supply chain. This lack of visibility is a critical security gap that requires specialized tools like Software Bill of Materials (SBOMs) and software composition analysis (SCA) to address.

THE LEGAL LABYRINTH

Navigating License Compliance

Beyond the technical vulnerabilities, the use of Open Source Software introduces significant legal and compliance risks. The OSS ecosystem is governed by a complex web of licenses, each with its own set of obligations, restrictions, and conditions. Failure to adhere to these licenses can result in severe consequences, including costly litigation, loss of intellectual property, and reputational damage.

The Landscape of Non-Compliance

Despite the high stakes, license compliance remains a widespread problem. The 2025 Black Duck OSSRA report revealed a startling level of non-compliance and risk exposure within commercial applications [2]:

- **56%** of the scanned codebases contained OSS license conflicts, typically between a permissive license and a strong copyleft license like the GPL.
- **33%** of the codebases contained open source components with no discernible license, placing the user in a state of legal ambiguity.

A separate 2019 whitepaper from Synopsys found that **85%** of audited codebases contained license compliance issues [5]. This high rate of non-compliance suggests a systemic failure in many organizations to properly track and manage their OSS obligations.

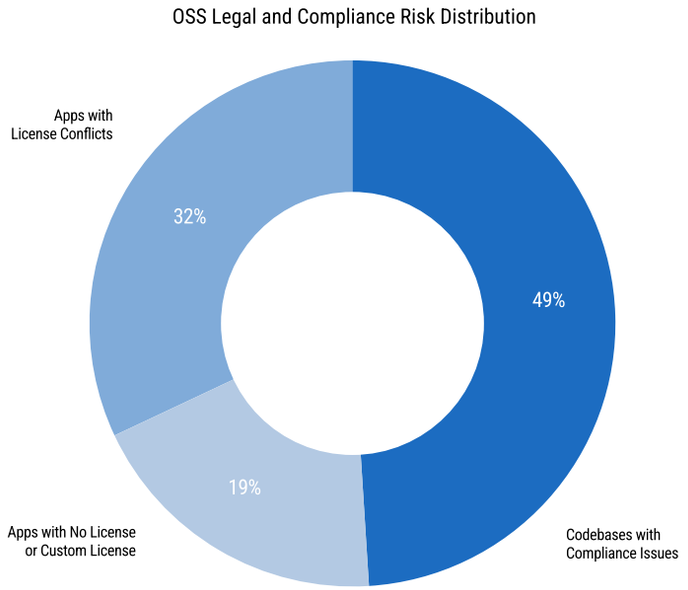


Figure 5: A significant portion of applications contain license conflicts or components with no license at all, creating a landscape fraught with legal risk. Data Source: Black Duck OSSRA 2025, Synopsys 2019

The High Cost of Legal Failure

The consequences of non-compliance are not merely theoretical. A growing body of case law demonstrates that courts are willing to enforce the terms of open source licenses, often with significant financial penalties. Analysis of major OSS license compliance lawsuits reveals a clear trend of escalating penalties [4].

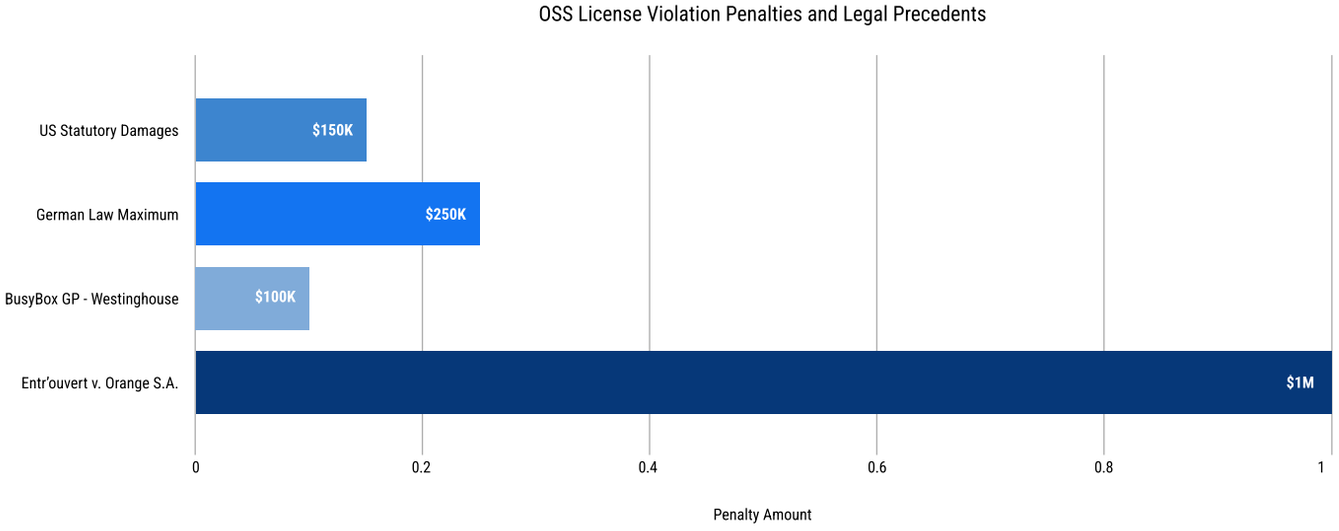


Figure 6: Major legal cases and statutory limits demonstrate that OSS license violations can result in penalties ranging from hundreds of thousands to over a million dollars. Data Source: FOSSA, 2025

Key legal precedents include:

- **Jacobsen v. Katzer (2006):** This landmark case established that OSS licenses are legally enforceable contracts and that violating their terms constitutes copyright infringement.
- **BusyBox GPL Enforcement (2007-2009):** A series of lawsuits by the Software Freedom Law Center resulted in multiple settlements, with Westinghouse paying over $100,000 in damages and costs and being forced to appoint an Open Source Compliance Officer.
- **Entr'ouvert v. Orange S.A. (2011-2024):** In a record-setting judgment, the Paris Court of Appeal ordered Orange to pay over €860,000 (>$1,000,000) for violating the GPL v2 license of the Lasso software. The damages were calculated based on the commercial license fee Orange had declined to pay.

The potential for consumer-led lawsuits, as seen in the ongoing *SFC v. Vizio* case, could further amplify the legal risks, creating a scenario where any end-user could potentially sue for non-compliance with GPL obligations [4].

BEYOND "FREE"

Uncovering the Total Cost of Ownership (TCO)

The most pervasive myth about Open Source Software is that it is "free." While the acquisition cost is typically zero, the Total Cost of Ownership (TCO) is far from it. A 2022 guide by the Qt Company, based on 15 years of commercial experience, provides a detailed model for calculating these hidden costs, which fall into several key categories [5].

Deconstructing the Hidden Costs

Using a real-world example of a 3-million-line-of-code framework, the analysis breaks down the annual recurring costs that organizations must account for when using OSS.

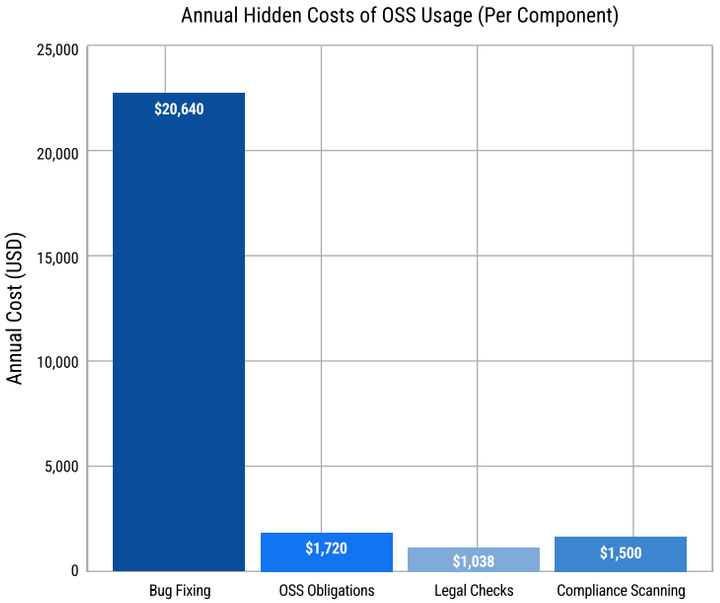


Figure 7: The annual Total Cost of Ownership for a single major OSS component can be substantial, with bug fixing and maintenance constituting the largest shares.
Data Source: Qt Company, 2022

The primary cost drivers are:

1. **Bug Fixing ($20,640/year):** Even assuming the community fixes 99% of bugs, the cost for an organization to fix the remaining 1% is significant. This calculation is based on an industry average of 20 bugs per 1,000 lines of code and an average developer cost of $688 per day.
2. **OSS Obligations ($1,720/year):** This includes the recurring effort to maintain user-facing lists of OSS components, handle source code requests, and update internal documentation.
3. **Legal Checks ($1,038/year):** The annual cost of having a senior attorney review new or updated licenses.
4. **Compliance Scanning ($1,500/year):** The subscription cost for automated tools (e.g., Snyk, Debricked) to scan for license and vulnerability issues.

Over a five-year period, the TCO for a single major open source component, including one-time setup costs of approximately $11,000, is estimated to be **$135,498** [5]. For organizations that rely on hundreds of OSS components, this figure can quickly multiply, representing a substantial and often unbudgeted operational expense.

The Risk of Unfunded Maintenance

These costs are predicated on the assumption that the open source project is actively maintained. However, many critical projects are supported by a small number of volunteers. When these maintainers burn out or abandon a project, the maintenance burden—and its associated costs—can shift entirely to the user. In such scenarios, the organization is left with several unpalatable options:

- **Fork the project:** Take over maintenance internally, incurring the full cost of bug fixing and security patching.
- **Migrate to a new component:** A costly and time-consuming engineering effort.
- **Accept the risk:** Continue using the unmaintained component, exposing the organization to unpatched vulnerabilities.

The TCO analysis reveals that OSS is not a free lunch. It is a strategic sourcing decision that requires a long-term commitment to resource allocation for maintenance, compliance, and risk management.

SOFTWARE SUPPLY CHAIN ATTACKS

The Evolving Threat Landscape

Perhaps the most alarming and rapidly evolving risk is the deliberate targeting of the open source software supply chain itself. Malicious actors, including sophisticated nation-state groups, have recognized that compromising a single popular OSS package can provide a backdoor into thousands of organizations. The 2024 Sonatype State of the Software Supply Chain report, which looks back at 10 years of data, chronicles the escalating sophistication of these attacks [6].

A Timeline of Escalating Sophistication

The nature of supply chain attacks has evolved from opportunistic exploits to highly targeted, long-term campaigns.

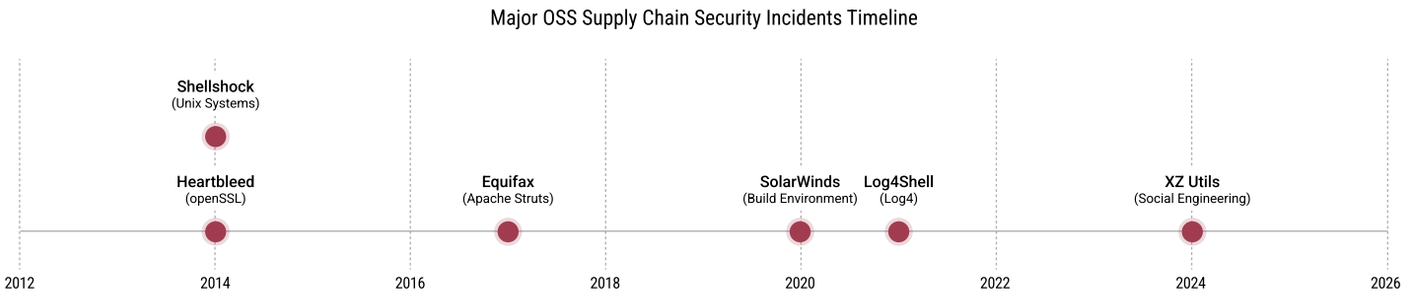


Figure 8: The timeline of major incidents reveals a clear trend towards more sophisticated and patient attacks on the OSS supply chain. Data Source: Sonatype, 2024

As the timeline shows, each major incident marked a step up in sophistication. The progression can be broken down as follows:

- **Early Years (2014-2017):** Attacks like Heartbleed, Shellshock, and the initial Apache Struts vulnerabilities were primarily opportunistic, exploiting known flaws in widely used components.
- **Targeted Malware (2017):** The first instances of attackers intentionally injecting malicious code into popular npm packages marked a shift towards targeted attacks.
- **Build Environment Compromise (2020):** The SolarWinds attack demonstrated a new level of sophistication, where attackers infiltrated the build process of a trusted vendor to distribute malware.
- **Ubiquitous Component Exploitation (2021):** The Log4Shell vulnerability showed how a flaw in a single, ubiquitous logging utility could set the internet "on fire," with exploitation beginning within hours of disclosure.
- **Social Engineering and Maintainer Takeover (2024):** The attempted XZ-Utils backdoor represents the current apex of supply chain attacks. This two-year-long operation involved patient social engineering to gain the trust of a lone maintainer, with the goal of embedding a backdoor into a fundamental Linux utility.

Since 2019, over **700,000 malicious packages** have been discovered, and the number of attacks **doubled again in 2024**, indicating that this is a rapidly growing threat vector [6].

The New Frontier of Risk

The XZ-Utils incident highlights a new and deeply concerning frontier of risk:

- **The "Benevolent Stranger" Playbook:** Attackers are no longer just exploiting technical flaws but are now exploiting the social fabric of the open source community—trust, collaboration, and the assumption of good faith.
- **Targeting Under-Resourced Projects:** Critical infrastructure often rests on projects maintained by one or two overworked and under-supported individuals, making them prime targets for manipulation and takeover.
- **Patience and Stealth:** Nation-state actors are willing to play the long game, spending years building credibility before executing their attack, making detection incredibly difficult.

This evolution of threats requires a paradigm shift in security, moving beyond reactive vulnerability scanning to proactive threat intelligence, behavioral analysis of code contributions, and a greater investment in supporting the maintainers of critical open source projects.

CONCLUSION

Towards a Balanced Risk Mitigation Strategy

The evidence presented in this article demonstrates that while Open Source Software is an engine of unprecedented economic value and innovation, it is accompanied by a spectrum of profound and often underestimated risks. The allure of "free" software has created a digital infrastructure that is simultaneously invaluable and fragile, built on a foundation where the incentives for consumption far outweigh the incentives for maintenance and security.

Our analysis has quantified the multifaceted nature of these risks:

- The **valuation paradox** creates systemic risk by concentrating immense dependency on a small, under-resourced group of maintainers.
- **Security vulnerabilities** are not the exception but the norm, with 86% of applications containing known flaws, a problem compounded by poor maintenance and the hidden danger of transitive dependencies.
- The **legal landscape** is a minefield of complex licenses, where non-compliance is rampant and can lead to million-dollar penalties.
- The **Total Cost of Ownership** is substantial, with hidden costs for bug fixing, compliance, and legal oversight often exceeding the price of commercial alternatives over the long term.
- The **threat of supply chain attacks** is escalating, with malicious actors employing increasingly sophisticated social engineering tactics to compromise the very heart of the ecosystem.

Addressing these challenges does not mean abandoning open source. That would be akin to abandoning the internet itself. Instead, it requires a fundamental shift from a model of passive consumption to one of active, risk-aware partnership. Organizations that build their businesses on OSS must adopt a holistic risk mitigation strategy that includes:

1. **Comprehensive Inventory and Visibility:** Implementing robust Software Composition Analysis (SCA) and maintaining a dynamic Software Bill of Materials (SBOM) to gain full visibility into both direct and transitive dependencies.
2. **Proactive Vulnerability and Patch Management:** Moving beyond reactive scanning to establish automated processes for identifying, prioritizing, and rapidly applying patches for known vulnerabilities.
3. **Rigorous License Compliance:** Integrating automated license scanning into the development lifecycle and establishing clear policies and legal oversight for OSS adoption.
4. **Budgeting for Total Cost of Ownership:** Recognizing that OSS is not free and allocating dedicated resources for maintenance, support, and the potential need to fork or migrate from unmaintained components.
5. **Investing in the Ecosystem:** Contributing back to the community, both financially and through developer time, to support the critical projects upon which the organization depends. This is not charity; it is a vital investment in the stability of one's own supply chain.

Ultimately, the long-term sustainability of the open source model depends on a shared responsibility between creators and consumers. By embracing a more mature and proactive approach to risk management, organizations can continue to harness the transformative power of Open Source Software while securing themselves against its latent and evolving dangers.

LOOK BENEATH THE SURFACE

Ready to Reveal What's Hidden in Your Open Source Stack?

The visible portion of your open source stack—what developers interact with daily—is only a fraction of the full risk landscape. Beneath the surface lie hundreds of transitive dependencies, outdated components, unresolved vulnerabilities, license conflicts, and operational obligations that shape your true exposure.

Quandary Peak Research helps organizations chart these hidden layers. Our assessments combine deep technical analysis with legal and security expertise to give you a complete picture of your OSS ecosystem and the confidence to manage it proactively.

If you'd like to see our process in action or gain access to a trial version of our technical due diligence platform, we'd be happy to set up a personalized demo.

Email us at info@quandarypeak.com to schedule your session and discover how our experts can help you make more confident, informed technology investments.

References

1 World Economic Forum (2025, February 3). Industrial Cyber.

WEF sounds alarm on software supply chain vulnerabilities, flags risks in open-source and third-party dependencies

https://industrialcyber.co/supply-chain-security/wef-sounds-alarm-on-software-supply-chain-vulnerabilities-flags-risks-in-open-source-and-third-party-dependencies/

2 Synopsys (2025). Black Duck.

2025 Open Source Security and Risk Analysis (OSSRA)

https://www.blackduck.com/blog/open-source-trends-ossra-report.html

3 Hoffmann, M., Nagle, F., & Zhou, Y. (2024). SSRN.

The Orbit of Open Source: The Economic Value of Unpaid Work

https://papers.ssrn.com/sol3/papers.cfm?abstract_id=4693148

4 FOSSA. (2025).

Analyzing 5 Major OSS License Compliance Lawsuits

https://fossa.com/blog/analyzing-5-major-oss-license-compliance-lawsuits/

5 Qt Company. (2022, May 10).

Guide to the Total Cost of Ownership of Open-Source Software

https://www.qt.io/blog/is-open-source-really-free

6 Sonatype. (2024).

2024 State of the Software Supply Chain Report: 10 Year Look Back

https://www.sonatype.com/state-of-the-software-supply-chain/2024/10-year-look

7 Lineaje. (n.d.).

Vulnerabilities by Dependency Level in Open-Source Projects

https://www.lineaje.com/chart-of-the-week/vulnerabilities-by-dependency-level-in-open-source-projects